ABSTRACT
Any actual status of task progress in an enterprise project is important facts for evaluating and controlling the project. Project manager can use for controlling cost, time, or quality as long as the project progress. To record the progress status of the projects based on using every new distributed application may be produce a difference format of data. It will make difficult for project manager to automatically generate the progress reports of projects or project performance. An alternative of independent and distributed intelligent application can be build to collect any various data format.

In this paper, we propose a part of our study in evaluating and developing a distributed application model based on mobile agent computing approach. A prototype of mobile agents has been developed by combining some mobile agent tools. In our model, mobile agents can be used to access and collect autonomously the actual information progress of enterprise projects for any running project on any where project server location. By some sample simulations, our mobile agents have been enabled to clone it self and travel for one host to any host in recognizing and collecting autonomously the progress status of projects.

Keywords: mobile agents, distributed application, enterprise projects, project performance, progress status of projects.

1. INTRODUCTION
Managing enterprise projects for most modern companies is now dramatically different from managing them at the departmental level. Bonham [7], in enterprise level might execute with portfolios of hundreds of projects and teams that are distributed geographically and even include external partners, outsourcers and offshore companies. Most of enterprise projects are no longer staffed by dedicated teams. These projects may be part of a larger, interdependent portfolio of projects with shared resources and common strategic goals. Thus with their complexities, interdependencies, resource constraints and time limits, projects are costlier, riskier and more mission-critical than ever before [6,32].

If an organization hopes to achieve its goals and succeed projects optimally, project manager that focuses only for best planning and running is not enough. In other word, Dinsmore [13] state that once a project begins, assuming that everything will go according to plan is a very dangerous strategy. And when projects is running, every task activity of projects must be properly is tracked, understood and administered over time in order to achieve its stated goals. In context process monitoring projects in distributed way or heterogeneous network and application become more complicated.

These situations have brought about a requirement for project management support systems to address the needs of distributed monitoring and controlling of projects. Software tools that make this possible have been done widely, both as independent parts or integrated fully applications. But, most of them are still using client-server approach, relatively needed much more cost, expensive and not easy to use. Furthermore, Schatten et al [35] these systems also give short of supporting the project manager in their decision making processes and do not offer assistance in representing knowledge about plans and designs, or provide mechanisms for reasoning about plans and designs in flexible ways.

The computing model base on mobile agents becomes an alternative technique and a very effective way for the development the future distributed application. As state by Jain et al [19] and Mangalwede et al [25] this model can be considered an additional design paradigm in the area of distributed programming and a useful supplement of traditional client-server techniques. Mangalwede et al [25], Kinshuk and Lin [22] also said that mobile agent technology can be viewed as a replacement, refinement or extension of the traditional client/server paradigm. The mobile agent framework emerged in the pursuit of open and decentralized models, relevant to the dynamic and distributed nature of computations on the internet.
Mobile agent technology has been used for much area of distributed applications, including for electronic commerce, monitoring and control of network server, system administration and management, and information retrieval [21,27,28,31,33,34].

In this paper, is present a part of our study in evaluating and developing of our model mobile agents for recognizing and collecting any progress data of enterprise projects in every location of company project servers.

2. RELATED WORKS

Development distributed software applications by using intelligent techniques to support managing business process in distributed location simultaneously have become a very interesting area of research for last few years. It also included the research of using the intelligent agent system and mobile agent approach for managing the new concept of project management in the business companies.

A number of attempts have been purposed to take advantage of mobile agents and multiagent systems approach in the field of distributed intelligent enterprise computing, for example see [5,11,14,30]. Most of them were also applied to monitor and control of enterprise networks performances, such as by Makki et al [24], and Abar et al [1]. Chuang and Chang [12] investigated the applicability of mobile agent for monitoring performance of the internet web server. In this research, network bandwidth was tested by purposefully record the network status of the agent server. Sadiig [33] showed a using of autonomous mobile agent-based distributed learning which delivered of the learning contents to their intended audience over a network. This distributed computational model is a very effectively factor for high-performance distributed computation.

In the field of project management, a few papers of using intelligent agent have been founded in literature. Such as for planning and risk management [18], management support interactively [20], scheduling [37], emotional model [26], and controlling [15]. O’Conner and Jenkins [29] and Andjomshoa et al [2] explored that artificial intelligence techniques could take account of the emerging requirement for support systems to address the increasing trend towards distributed multi-platform software development projects. For example, Barber [4] showed that reasoning techniques can be used to track the mismatch between specifications and implementations of projects. In other words, analogy techniques can be used to look for existing specifications, components or implementations that match some new requirement.

O’Conner and Jenkins [29] also stated that a software system can be designed to act as an intelligent team member, which could help in the planning and execution of a project. This intelligent project assistant could help to preserve knowledge about tasks, to record the reasons for decisions and retrieve information relevant to new problems. O’Connor and Jenkins [29] also stated that an intelligent project assistant as a useful tool for all project managers. Because, it could intelligently manage project knowledge, capture knowledge and lessons learned about projects into a project knowledge base.

Another reason why in distributed business management environment needs to use intelligent agents is said by Houari and Far [16]. They argued that agents could support distinct individuals of organization in decision making process as a natural way of representing the broadly categorized business users of an organization. In their research, each autonomous agent could delegated to from the business user, controlled over its actions, used its own knowledge capabilities, adapts, interacts, and responded to changes in its environments.

A good work of using mobile agent paradigm in a monitoring and control aspect for virtual enterprise collaboration was shown by Bergamaschi et al [6]. In this paper, they designed the new behavior and code of conduct mobile agent web, such as the willingness to share data and knowledge, in order to support mediate negotiations between the parties of virtual enterprise.

In paper [14,21], the authors realized that they could actually apply mobile agent technology to collect price products. It sends out agents to different hosts in an electronic marketplace. Then the agents collect and report information such as prices and availabilities about products specified by users. In simple words, the system can collect and compare the prices of a set of products specified by users from different seller hosts in an electronic market.

A model of mobile agent which able to deliver newly updated data can also be built, such as is proposed by Armo et al [3]. An intelligent mobile agent was designed to get updated info, get access to latest scores, created daily Quiz competitions about the recent local football happenings. Then the mobile agent delivered newly updated data to the Mauritian population.

To access data in distributed resource location by a mobile agent could be formatted by some ways, such as RDF metadata model [2,35] Short Message Service [3]. Andjomshoa [2], and Schatten et al [35]
had developed RDF abstract model for annotating existing data projects in each resource center, in order to make these distributed resources available to remote users. A mobile agent is sent for responsibility of sending knowledge mining and filtering the RDF data. While Armo et al [3], information was successfully updated via Short Message Service and synchronization of data on the mobile device with data on the desktop computer or backend server and vice versa is implemented successfully.

3. MOBILE AGENT

A mobile agent application can be viewed as a proactive computer program which can travel from one location to other location in computer networks [8]. In Figure 1 is shown an illustration of fundamental characteristic different between client-server model and mobile agent computing. In client-server computing, there is only data message which is sent between hosts Figure 1(a), while in mobile agent computing both program code and data message can be sent Figure 1(b).

Beside that, a number of artificial capability modules can also be embedded into these mobile agents in order to work or act naturally as human behaviors. These abilities can be chose to be activated every time by agent itself without direct participation of the users. As simply stated by Bellavista [5], mobile agents are mobile, flexible, autonomous, dynamic and efficient. When encapsulated within a task, a mobile agent can be dispatched to a remote host by the original host. After executing and accomplishing its tasks at the remote host, it can bring the results back by returning to the original host or send them through a message.

Chuang and Chang [12], a mobile agent, as a moving software agent, can be consisted generally of the following components. As illustrated in Figure 2.

![Figure 2. An internal structure of mobile agent](image)

**Itinerary**, this is the place where the route of the agent is recorded. Not only does it record the route, it also records the current position, so that we know exactly where the agent is. This is an essential part of a mobile agent; since the agent is moving, it needs to know where to start and where to go next and ultimately, where to end. **Code**, this is the part where fragments of the program code are stored. By definition, an agent is a computational entity, which also means it is a program that can be executed. Without this code part, an agent will not be able to perform any tasks, not even traveling. Moreover, this part controls all other parts of the agent.

**State**, status of the agent is recorded here. In our case it means the status of the tested hyperlinks will be recorded here. With this part, the agent then will be able to report to both the server and client about the status of the hyperlinks. **Host**, this is where the server position is stored. It is quite vital since an agent is only running on the Agent Transfer Protocol (ATP). The agent has to remember where it came from so that it will be able to return to the server after the tasks assigned are completed. If not, it will be lost in the network and perhaps jam up the network. **Other necessary details** – The agent needs to show who created it, and this is the place to put that. Other information related to this agent is also stored here so that people will know what this agent does and who the owner is.

The tasks can also be decomposed and encapsulated into multiple mobile agents. Every mobile agent can run independently to accomplish its task. Thus, a set of mobile agents can run in parallel on distributed hosts so that the whole task can be completed in a shorter time. Additional, both on client and server computer, mobile agents can change locations according to the available network resources. In a typical scenario, an agent starts on
the client side, moves to the server that provides resources, performs operations on the server side (e.g., information retrieval and filtering) and then returns the final results to the client side [5].

Another capability of mobile agents is enable to clone itself. According to Lam et al [22] and Shehory et al [35], cloning is a possible response of an agent to overloads. When agent overloads are due, in general, either to the agent's limited capacity to process current tasks or to machine overloads. Some condition an agent should consider in cloning are: 1) If cannot perform all of its tasks on time by itself or decompose them so that they can be delegated to others. 2) There is no lightly loaded agent that can receive and perform its excess tasks (or subtasks when tasks are decomposable). 3) There are sufficient resources for creating and activating a clone agent (on either the same machine or a remote one). 4) The efficiency of the clone agent and the original agent is expected to be greater than that of the original agent alone.

In Figure 3 is shown a mechanism of cloning an agent. When such information is not available, an agent may compute the expectation values of the attributes of remote machines relying on probability distributions either specifically by machine ID (e.g., IP address) or group wise by machine type.

![Figure 3. A Mechanism of agent cloning model](image)

In order to perform useful tasks, mobile agents should be able to communicate with each other. Communication is an essential ability for mobile agents to collaborate with others by information exchanging and knowledge sharing. According Coa et al [10], each mobile agent should use the same transport protocol to understand the message transport protocol; same agent communication language (such as ontology, KQML, FIPA ACL); and if agents are using some interaction protocol then must be informed specifically.

One of the main differences between mobile code, such as applets, and mobile agents is itinerary. Whereas mobile code usually travels just from point A to point B, mobile agents have an itinerary and can travel sequentially to many sites. One natural application of mobile agents, therefore, is collecting information spread across many computers hooked to a network [5].

The computing model with mobile agents technology has numerous attractive aspects and advantages, including: a) can reduce network traffic by performing time and traffic consuming operations locally on the server; b) can continue functioning even if its home is unavailable or unreachable, and send back the results upon reconnection; c) can be flexibly and dynamically deployed of real time software components that in the form mobile agents [11,19,22,24].

4. EARNED VALUE METHOD

Every early time executes the enterprise project plans, a professional project manager must always provide monitoring and evaluation of project progress to control periodically and keep the project on track. Such as technical issues, illness, project scope, personnel changes, time of task completion, or budget reduction.

According to Howes [17], the project manager must have a proactive toolset that allows ongoing monitoring of a wide range of project parameters (budget, schedule status, resources status, etc.) in order to spot problems before they become crises. To support this situation, company needs a robust solution system that integrates project management, project performance and process control. Such that allow team, staff to collaborate on project management and provides real-time visibility into project status.

Boham [7], the Earned Value Method (EVM) is a good tool that provides measures to evaluate cumulatively any running projects. The measurement is calculated easily in percent complete for specific tasks, plot manpower loading in S-curves. EVA can be used to develop an early warning system while track weekly and job performance factors. Common ways to calculate the earned value can be written as equations (1), (2).

\[
\text{Earned Value} = \text{Qty Installed } \times \text{Estimated Unit Rate} \quad (1) \\
\text{Earned Value} = \text{Budgeted Hrs } \times \% \text{ Complete} \quad (2)
\]

A definition of earned value is basically allows a project manager office to control overall project cost and total percentage of complete for unlike work tasks. For example equipment installation and LF of wiring integrates cost and schedule tracking and control, can be calculated by equations (3), (4).

\[
\text{Job } \% \text{ Complete} = \frac{\Sigma \text{Earned Hours}}{\Sigma \text{Base Hours}} \quad (3) \\
\text{Performance Factor} = \frac{\Sigma \text{Earned Hours}}{\Sigma \text{Actual Hours}} \quad (4)
\]

And the current percentage of complete can be used to forecast total work hours at project completion, using equation (5).

\[
\text{Forecast} = \text{Work hours to date } + \% \text{ Complete} \quad (5)
\]
This time tracking result, from the company perspective is ability to manage resource utilization, improve estimation by building historical estimation database, improve visibility into schedule and budget, and drive repeatability and consistency.

5. SYSTEM ARCHITECTURE

Our mobile agent application is designed by combining of agent migration and remote communication, using plain sockets. In Figure 4, is shown the overall diagram of our mobile agent model. There are three main basic types of mobile agents: Collection-agent (CA), Report Agents (RA), and User-agent (UA).

Collection agents reside on the master server site. Travel to any department server site of enterprise projects and clone themselves for recognizing, accessing and collecting progress data of unit project tasks. Such as actual cost of works, actual time of task completion. Report agents reside on the master server site, manage and control the collection of data from Collection agents, prepare resume of enterprise progress report and travel around to user client sites. User-agents provide the basic user interface of the application, close to the end-user, request and receive processed data from Report agents and visualize it relevantly.

6. COMMUNICATION PROCESS

In this research, we design each mobile agent can only communicate with other mobile agents at the host their currently resides. Both a message passing and a mailbox strategy can be used by agents that are formed in KQML and FIPA as agent communication languages.

With the message passing we mean that all the messages are sent to the agent home and the agent receives messages from its home. While with the mailbox (logically one part of agent) migrates with the agent all the time. The mailbox can be detached from its owner to a new host while leaving its mailbox at a previous host along its migration path. Thus the communication process is done by transmitting messages from the sender to the receiver’s mailbox then delivering the messages from the mailbox to its owner agent.

Every mobile agent in the system is allocated a mailbox. A mailbox is a message buffer used to store incoming messages. Incoming messages sent to the agent are inserted into the mailbox. As shown in Figure 5, if every time a cloning collection agent recognizes a new update status of project tasks then cloning agent send its to collection agent by simply sends the message to the receiver’s mailbox. The receiver of collection agent receives the message from its mailbox. The collection agent then asks immediately the report agent to update the project performance and reporting.

7. RESULTS AND DISCUSSION

We have implemented a prototype system by using some java agent tools for a distributed environment. The tools also have the advantages of platform independence, good security mechanism and easy of programming. Our prototype system includes the core agent system to make up of multiply mobile agents and graphical user interfaces, which provides the possibility to control the configuration of the each mobile agent behaviors.

To measure performance of our mobile agent application prototype we simulate experimentally some project cases for monitoring and control from
a telecommunication company. There are a lot of projects that must be executed dynamically. Such as to install, construct, or assemble of stakes, cables, telephones, internet, contents services, etc. In our prototype, we assume that the master plan of all enterprise projects have been recorded in department servers. A generator to generate continually of any information of project status we put in each department server. These generators can be viewed as replacement of any project technical operator or the person from a team project which is updating their current status progress of activities.

Collection mobile agent is sent out to trace down the project status on each department server. As we said above, when collection agent arrives to any department server host then clones itself to start searching and collecting any current update data of project activities. At the same time, collection mobile agent continues traveling to other servers. After visiting all project servers the collection agent comes back to master project sever to wait and receive the information of new project progress status from every clone collection agent. Then the collection agent contacts and delivers continuously to report mobile agent in order to generate automatically enterprise project progress and performance.

Thus from the user interface, project manager can get an overview of the whole project and gather the status of the project. This interface can also be started as a stand-alone java program or embodied in a web page browsed by an internet browser. As users update their status, information is automatically rolled up through the project hierarchy so that project managers and stakeholders can see current project status in real time, all the time.

The report mobile agent uses these collecting progress project, reformat information into new status enterprise project reports. Then the report mobile agent evaluates and develops a simply recommendation for each enterprise projects. Project status benchmarks include graphical indicators as to how well the project is tracking according to schedule, scope, and budget objectives. This project status information is published centrally by report mobile agent, with the objective of keeping all stakeholders and dependent project teams informed.

From that some experimental simulation we can show that this our distributed mobile agent prototype could support inter-project dependencies, and updated continually in real time. So that the projects can go on external tasks and projects from pushing deadlines without stakeholders knowing about changes. Management can also receive some specific information in order to make decisions about reallocating budget and resources, adjusting schedules, and the performance of project teams.

8. CONCLUSIONS

Our mobile agent prototype has enabled to roam on the enterprise network to collect information about the status enterprise projects that are distributed on department project servers. These mobile agents can be execute jointly or separately in order to generate automatically the report progress and performances of enterprise projects.

The collection agents can be sent and collected information related to department project planning and progress status. Such as scheduling, budget, actual cost, task completion. Then at the same time, the report agent uses this information to make evaluation and generate enterprise project progress and performance. From simulation results show that the prototype could improve scalability and availability, provide fast and efficient interaction to any user remote.

REFERENCES


